



# CodeWarrior Development Studio for Power Architecture® Processors Version 10.x Quick Start

## SYSTEM REQUIREMENTS

**Hardware**

Intel® Pentium® 4 processor - 2 GHz or faster,  
Intel® Xeon™, Intel® Core™, AMD Athlon™ 64,  
AMD Opteron™, or higher  
2 GB RAM  
CD-ROM drive for CD installation  
Microsoft Mouse compliant pointing device  
Internet connectivity for Web downloads and  
update access

**Operating System**

Windows:

- Windows 7 SP1 (32-bit and 64-bit)
- Windows 10 (64-bit)

Linux:

- Ubuntu 14.04 (32-bit and 64-bit)
- CentOS 7.0.1503 (64-bit)
- OpenSUSE 13.2 (64-bit)
- Debian 8.2.0 (64-bit)
- Fedora 22 (64-bit)
- RHEL 6.5 (64-bit)
- Mint 15 (64-bit)

**NOTE:** Other Linux distributions can be used, but will likely require more manual identification and installation of missing, required libraries.

**Disk Space**

2.3 GB, additional space required during  
installation

This document explains how to install the CodeWarrior software and how to use the software to create, build, and debug a demonstration Power Architecture multi-core processor project.

... the procedures that follow, advanced users can use the numbered steps. Novices should use the more detailed instructions provided by the substeps.

## Section A: Installing and Registering CodeWarrior Software

### CodeWarrior installation on Windows OS

Installing CodeWarrior software on Microsoft Windows 7 operating system requires administrator rights, because the installer copies files into the `System` folder. The default CodeWarrior installation folder is `C:\Freescale\CW_PA_v10.x`.

In addition, your project workspace needs to be set up in any folder that you can fully access.

1. Insert the **CodeWarrior Development Studio** installation CD into the CD-ROM drive.

The CodeWarrior installation menu appears.

**NOTE** If autorun is disabled on your computer, click **Start > Run** and enter `cd_drive:\Launch.exe` where `cd_drive` is the drive letter assigned to the CD-ROM drive.

2. Run the installer.

The install wizard appears.

3. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

When installation completes, the **InstallShield Wizard Completed** page appears.

4. Click **Finish**.

You have successfully installed CodeWarrior for Power Architecture.

### CodeWarrior installation on Linux OS

Eclipse needs read/write access to the installation folder. Ensure that the eclipse installation folder has the appropriate permissions for all users.



your project workspace has read and write permissions. If the CodeWarrior software does not restart automatically after a successful CodeWarrior update operation, run `./cwide -clean` to launch the CodeWarrior software.

1. Insert the **CodeWarrior Development Studio** installation CD into the Linux host computer's CD-ROM drive.
2. On the host computer, open a new terminal window.  
A shell session starts.
3. Mount the CD-ROM media on the Linux file system.
4. Change the working directory to the CD-ROM mount directory.

---

**NOTE** See the `README.txt` file in the mount directory. This file contains installation instructions of different Linux distributions.

---

5. Issue the command: `xhost`
6. Issue the command: `./setuplinux`  
The install wizard starts and displays its welcome page.
7. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

When the software installation completes, the wizard displays its installation summary page.

8. Click **Finish**.  
This completes the installation of CodeWarrior software on the Linux machine.

---

**NOTE** To uninstall CodeWarrior from a Linux machine, open a terminal window, navigate to the CodeWarrior installation folder, run the `./uninstall` command from a root account, and follow the wizard's on-screen instructions.

---

---

**NOTE** For licensing and activation instructions for your product, see *CodeWarrior Development Suite Quick Start*. Save the license



, license.dat, to the installation directory. The default installation directory is C:\Freescale\CW\_PA\_V10.x\PA.

---

**NOTE** CodeWarrior service packs are installed with the Eclipse Updater. The updater must also be run using administrator rights. To start the Eclipse Updater, select **Help > Install New Software** from the CodeWarrior IDE menu bar.

---

1. Launch the CodeWarrior IDE.
  - a. **On Windows:**

Select **Start > All Programs > Freescale CodeWarrior > CW for Power Architecture v10.x > CodeWarrior IDE** from the Windows taskbar.
  - On Linux:**
    - Open a new terminal window and change the working directory to: `<CWInstallDir>/eclipse/` (where `<CWInstallDir>` stands for the directory in which you installed the CodeWarrior software).
    - Issue the command `./cwide`

---

**NOTE** On OpenSUSE 64-bit, run the `./cwide` command either as a normal user for regular privileges, or as a super user (using the `su` command) for escalated privileges. Normal escalation using the `sudo` command does not work correctly on OpenSUSE.

---

The **Workspace Launcher** dialog box appears.

- b. If you wish to change the location of your project's workspace, click **Browse** to select the new path.

The **Select Workspace Directory** dialog box appears.

- c. Select the required folder. Alternatively, to create a new workspace directory:
  - On Windows, click **Make New Folder**.
  - On Linux, click **Create Folder**.
- d. Click **OK**.

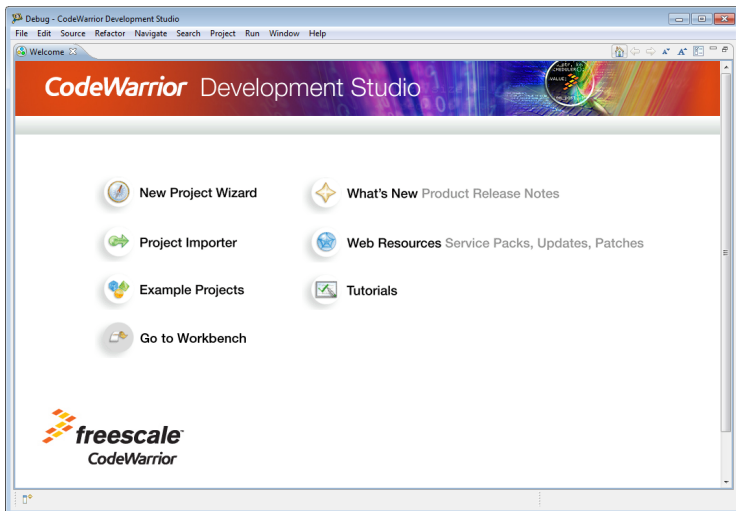
The **Select Workspace Directory** dialog box closes.

- e. Click **OK** to store the project at the specified location.
- f. CodeWarrior launches and displays the **Welcome** page.

---

**NOTE** The **Welcome** page is displayed when CodeWarrior is run for the first time. The **Welcome** page can be opened later by selecting **Help > Welcome** from the CodeWarrior IDE menu bar.

---

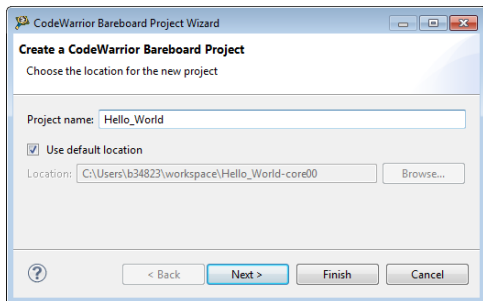



---

**NOTE** The **Project Importer** link on the **Welcome** page allows you to import only a CodeWarrior Classic project ( .mcp) file.

---

- g. Click **Go to Workbench** on the **Welcome** page.  
The workbench window appears.
2. Create a new project.
  - a. From CodeWarrior IDE menu bar, select **File > New > CodeWarrior Bareboard Project Wizard**.  
The **CodeWarrior Bareboard Project Wizard** launches and the **Create a CodeWarrior Bareboard Project** page appears.



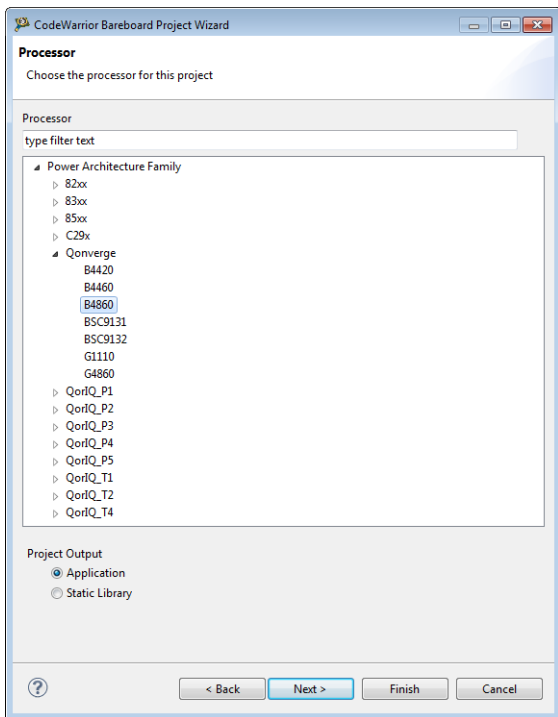
- b. In **Project name** text box, type `hello_world`.

---

**NOTE** The **Location** text box shows the default workspace location. To change this location, clear the **Use default location** check box and click **Browse** to select a new location.

---

- c. Click **Next**.  
The **Processor** page appears.

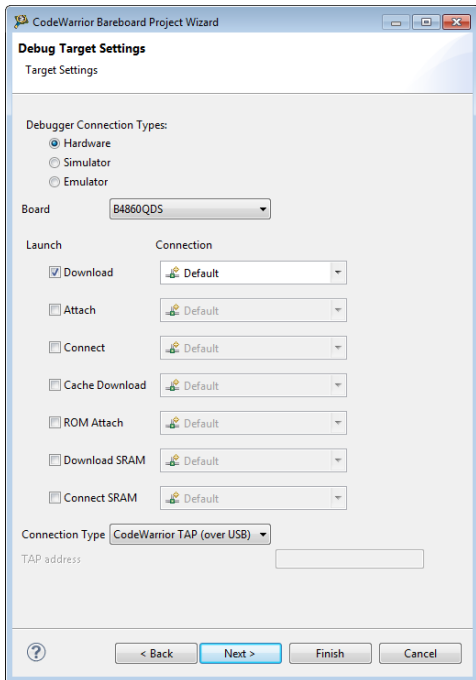


- d. Select a target processor for the new project, from the **Processor** list.
- e. Select **Application** from the **Project Output** group to create an application with `.elf` extension, which includes information required to debug the project.
- f. Click **Next**.  
The **Debug Target Settings** page appears.
- g. Select a connection type (hardware, simulator, or emulator), from the **Debugger Connection Types** group.
- h. Select the board you are targeting, from the **Board** drop-down list.



You can only select hardware or simulators that support the target processor selected on the **Processors** page. If you are using a Simics simulator, then see <http://www.windriver.com/products/simics/> for latest version and installation instructions for Simics simulator.

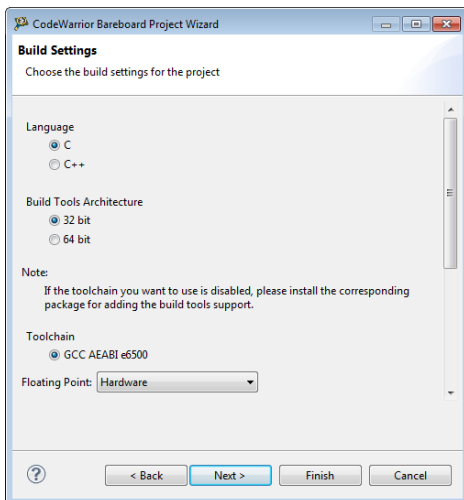
## Debug Target Settings Page



- i. Select the launch configurations and the corresponding connection to be included in your project, from the **Launch** group.
- j. Select the interface to communicate with the hardware, from the **Connection Type** drop-down list.
- k. Enter the IP address of the TAP device in the **TAP address** text box. This option is disabled and cannot be edited, if you select **USB TAP** from the **Connection Type** drop-down list.
- l. Click **Next**.

The **Build Settings** page appears.

## Build Settings Page

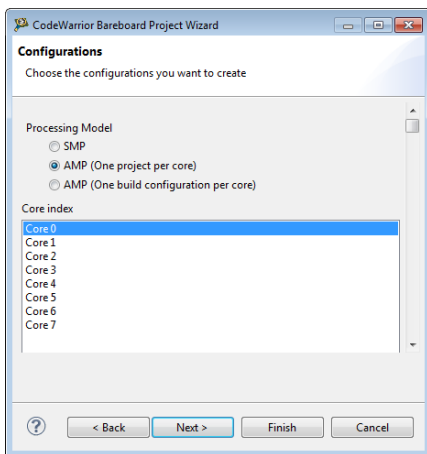


- m. Select a programming language, from the **Language** group.  
The language you select determines the libraries that are linked with your program and the contents of the `main` source file that the wizard generates.
- n. Select the processor used by the new project, from the **Build Tools Architecture** group.
- o. Select a toolchain from the **Toolchain** group.  
Selected toolchain sets up the default compiler, linker, and libraries used to build the new project. Each toolchain generates code targeted for a specific platform.

...the current release supports only a few selected toolchains. If the toolchain you want to use is disabled, you will have to install the corresponding service pack for adding the required build tools support. For more information, see *Service Pack Updater Quickstart.pdf* available in the `<CWInstallDir>\PA` folder.

- p. Select an option from the **Floating Point** drop-down list, to prompt the compiler to handle the floating-point operations by generating instructions for the selected floating-point unit.
- q. Click **Next**.  
The **Configurations** page appears.

### Configurations Page



- r. Select a processing model option from the **Processing Model** group.

**NOTE** The **SMP** option is available for selection only while creating projects for some e500mc, e5500, and e6500 core targets.

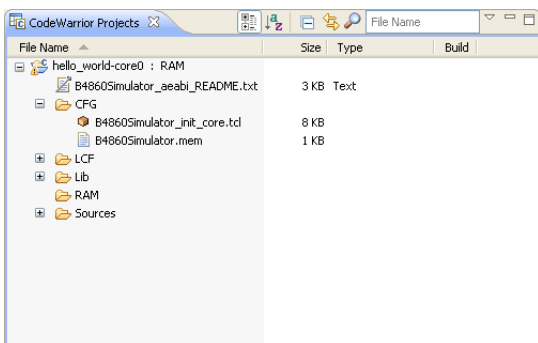
- s. Select the processor core that executes the project, from the **Core index** list.

debug programs on a multicore processor, you can select different core index for creating separate projects for each core.

- t. Click **Finish**.

The wizard creates a simulator project according to your specifications. You can access the project from the **CodeWarrior Projects** view on the workbench.

### CodeWarrior Projects View



### 3. Build the program

- a. Select the newly created project in the **CodeWarrior Projects** view.
- b. Select **Project > Build Project** to build the project. Alternatively, right-click the project in the **CodeWarrior Projects** view and select **Build Project** from the context menu that appears.

The IDE compiles the project's source code files and links resulting object code into an ELF-format executable file.

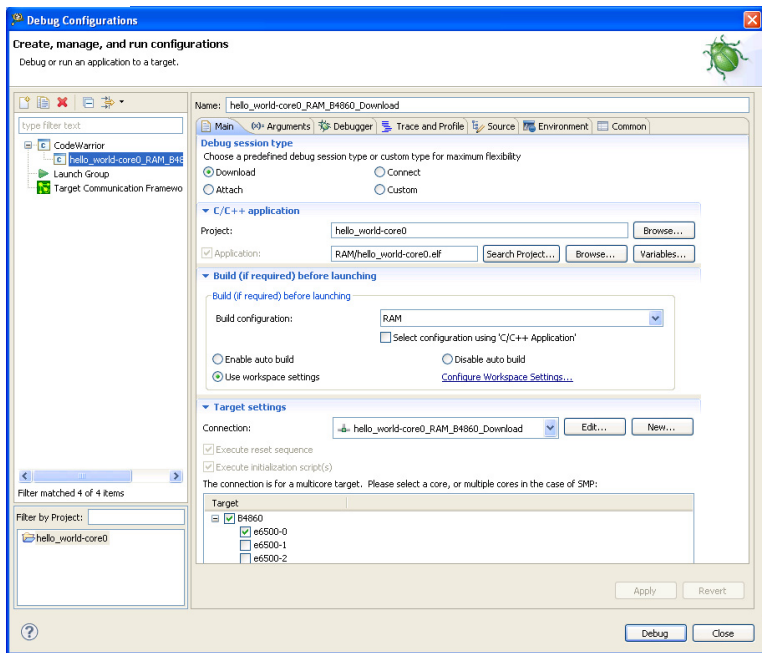
### 4. Prepare to debug the program

- a. From the CodeWarrior IDE menu bar, select **Run > Debug Configurations**.

The **Debug Configurations** dialog box appears.

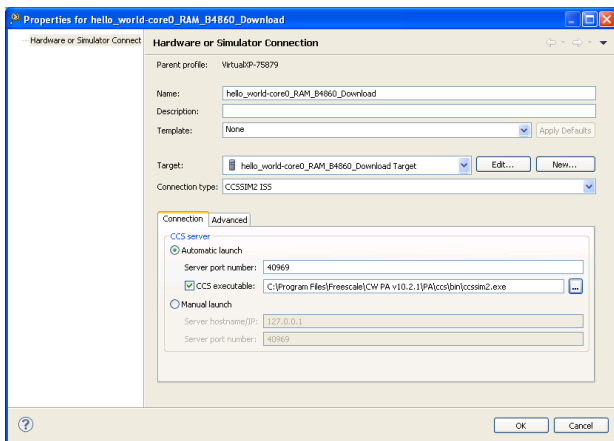
- b. In the left pane of this dialog box, expand the **CodeWarrior** group.
- c. Select the launch configuration for the newly created project.

A set of tabbed configuration panels appears in the right pane of the dialog box.



- d. In the **Main** tab page, select a remote system from the **Connection** drop-down list.
- e. Select a core, or multiple cores in case of SMP, from the **Target** list.
- f. Click **Edit**.  
The **Properties for <connection>** window appears.
- g. Select a connection type from the **Connection type** drop-down list.
- h. Configure the **CCS server** settings on the **Connection** tab.

## Properties for <connection> Window





- i. Click **OK**.  
The **Properties for <connection>** window closes.
  - j. Click **Apply**.  
The IDE saves your settings.
5. Debug the program.
- a. Click **Debug**.
  - b. The IDE switches to the **Debug** perspective. The debugger downloads your program to the target board and halts execution at the first statement of `main()`.

---

**NOTE** To download multiple projects on each core, you can click the pull-down menu next to the debug icon. From this menu, pick the next core you wish to debug.

---

- c. Click a thread in the **Debug** view.  
The program counter icon  (on the marker bar) points to the next statement to be executed.
  - d. In the **Debug** view, click **Step Over** .  
The debugger executes the current statement and halts at next statement.
6. Set breakpoint and execute program to breakpoint.




editor area, scroll to this statement

```
while (1) { i++; } // loop forever
```


- b. Double-click the marker bar next to the statement.

A breakpoint indicator  appears next to the statement.


- c. In the **Debug** view, click **Resume** .

The debugger executes all statements up to but not including the breakpoint statement.


## 7. Control the program:

- a. In the **Debug** view (top-left of perspective), click **Step Over** .

The debugger executes the current statement and halts at the next statement.

- b. In the **Debug** view, click **Resume** .

Execution resumes, the program writes the strings “Core0: Welcome to CodeWarrior!” and “Core0: InterruptHandler: 0xc00 exception.” to the terminal emulator, and then enters an infinite loop.

- c. In the **Debug** view, click **Terminate** .

The program terminates and the debug session ends.

## 8. Select **File > Exit**.

The CodeWarrior IDE window closes.

---

# Congratulations!

**You have created, built, and debugged a Power Architecture project using the CodeWarrior software.**

---



Freescale logo, CodeWarrior, QorIQ, StarCore are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. QorIQ Qonverge is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2009-2016 Freescale Semiconductor, Inc.

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale was negligent regarding the design or manufacture of the part.

## How to Contact Us

|                               |  |
|-------------------------------|--|
| <b>Corporate Headquarters</b> | Freescale Semiconductor, Inc.<br>6501 William Cannon Drive West<br>Austin, Texas 78735<br>U.S.A. |
| <b>World Wide Web</b>         | <a href="http://www.freescale.com/codewarrior">http://www.freescale.com/codewarrior</a>          |
| <b>Technical Support</b>      | <a href="http://www.freescale.com/support">http://www.freescale.com/support</a>                  |

Revised: 19 January 2016

